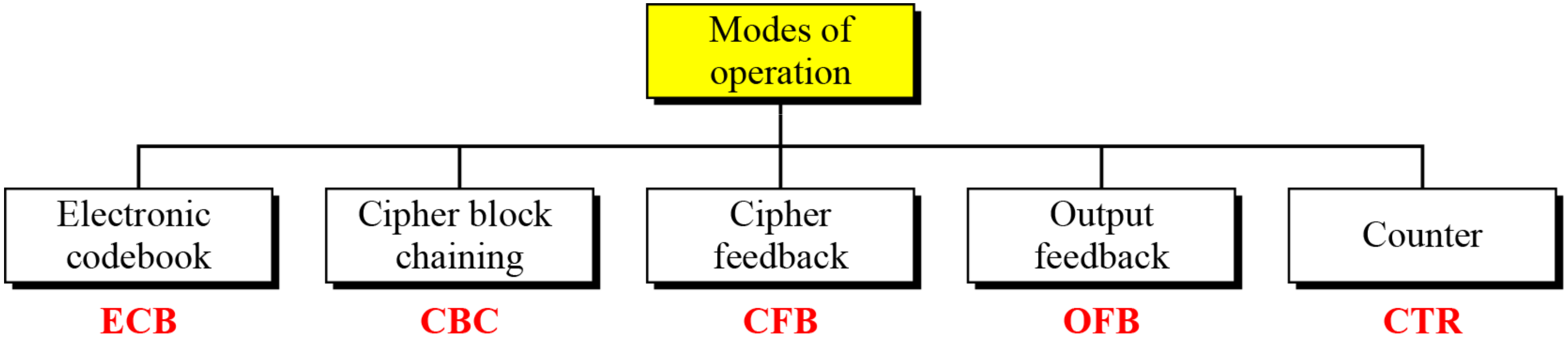
Block cipher modes of operation

[Slide courtesy: Cryptography and network security by Behrouz Fourozan]

Modes of operation

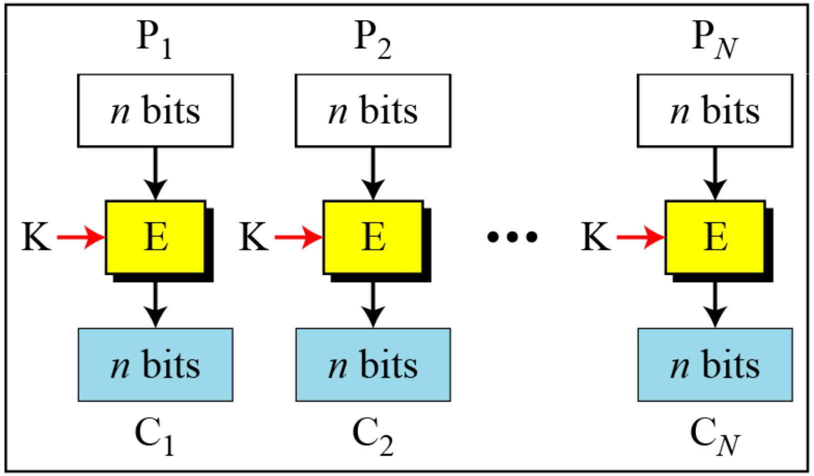
- Modes of operation have been devised to encipher text of any size employing either DES or AES.



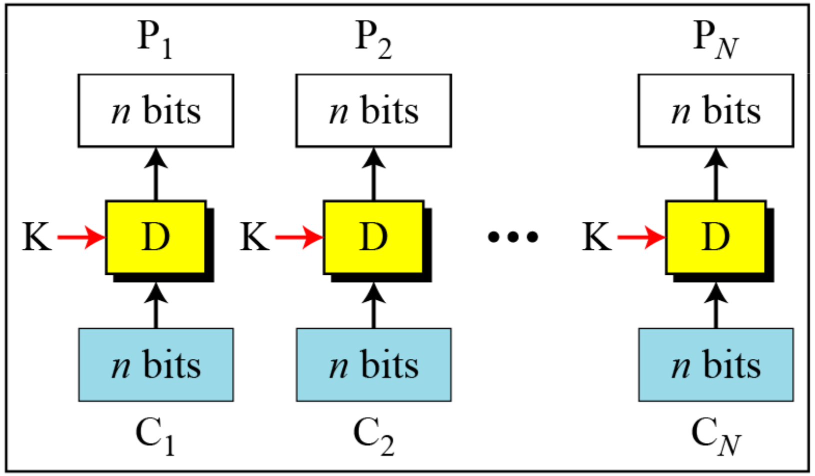
Electronic Codebook (ECB) Mode

Encryption: $C_i = E_K(P_i)$ Decryption: $P_i = D_K(C_i)$

E: Encryption D: Decryption
P_i: Plaintext block i C_i: Ciphertext block i
K: Secret key



Encryption



Decryption

Electronic Codebook (ECB) Mode...

- It can be proved that each plaintext block at Alice's site is exactly recovered at Bob's site.
  - Because encryption and decryption are inverses of each other,

Encryption: $C_i = E_K(P_i)$ Decryption: $P_i = D_K(C_i)$

- Called electronic codebook
  - because one can precompile 2^K codebooks (one for each key) in which each codebook has 2^n entries in two columns.
  - Each entry can list the plaintext and the corresponding ciphertext blocks.

Electronic Codebook (ECB) Mode...

- Assume that Eve works in a company a few hours per month (her monthly payment is very low). She knows that the company uses several blocks of information for each employee in which the seventh block is the amount of money to be deposited in the employee's account. Eve can intercept the ciphertext sent to the bank at the end of the month, replace the block with the information about her payment with a copy of the block with the information about the payment of a full-time colleague. Each month Eve can receive more money than she deserves.

Electronic Codebook (ECB) Mode…

- Error Propagation
  - A single bit error in transmission can create errors in several in the corresponding block.
  - However, the error does not have any effect on the other blocks.

Electronic Codebook (ECB) Mode…

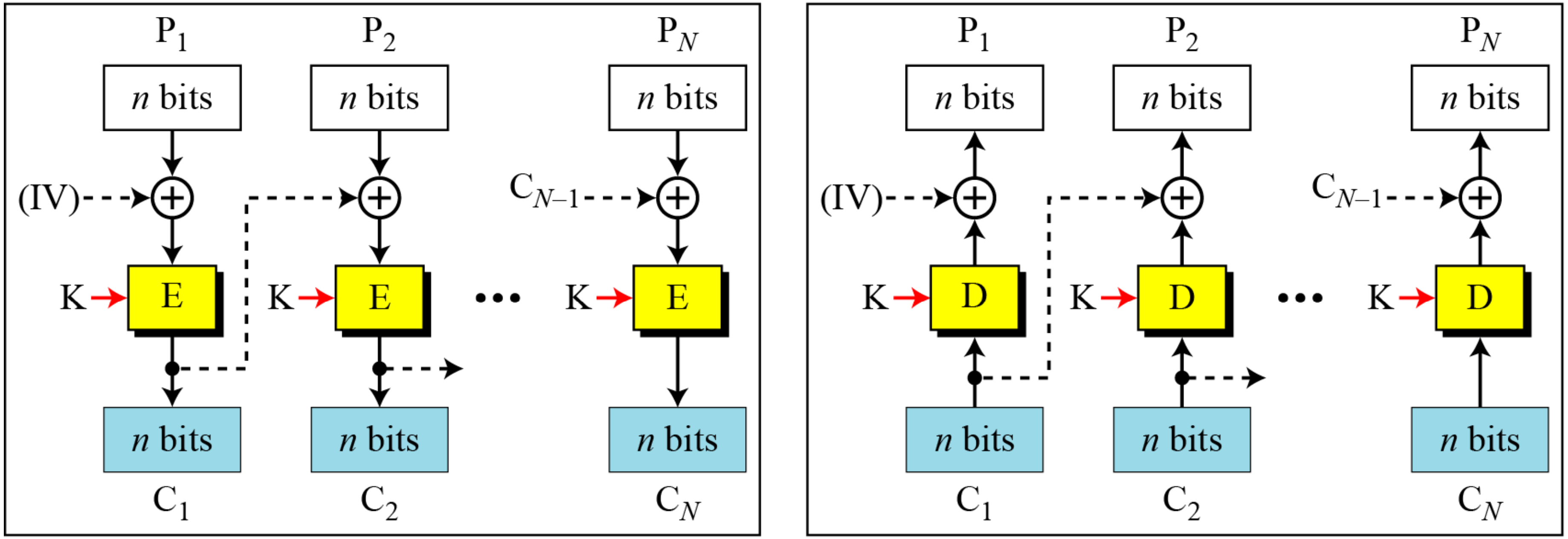
- What if one does not want to use padding?or the plaintext is fixed and stored somewhere?
  - A technique called ciphertext stealing (CTS) can make it possible to use ECB mode without padding.
  - In this technique the last two plaintext blocks, P_{N-1} and P_N , are encrypted differently and out of order, as shown below, assuming that P_{N-1} has n bits and P_N has m bits, where m ≤ n

$$
\begin{array}{lcl}
X = E_K(P_{N-1}) & \rightarrow & C_N = head_m(X) \\
Y = P_N \mid tail_{n-m}(X) & \rightarrow & C_{N-1} = E_K(Y)
\end{array}
$$

Cipher Block Chaining (CBC) Mode

- Each plaintext block is exclusive-ored with the previous ciphertext block before being encrypted.

E: Encryption | D : Decryption
P_i: Plaintext block *i* | C_i : Ciphertext block *i*
K: Secret key | IV: Initial vector (C_0)



Encryption Decryption

Cipher Block Chaining (CBC) Mode

- Prove that each plaintext block at Alice's site is recovered exactly at Bob's site.

Cipher Block Chaining (CBC) Mode

- Prove that each plaintext block at Alice's site is recovered exactly at Bob's site.
- Because encryption and decryption are inverses of each other,

$$P_i = D_K(C_i) \oplus C_{i-1} = D_K(E_K(P_i \oplus C_{i-1})) \oplus C_{i-1} = P_i \oplus C_{i-1} \oplus C_{i-1} = P_i$$

- Initialization Vector (IV)
  - The initialization vector (IV) should be known by the sender and the receiver.

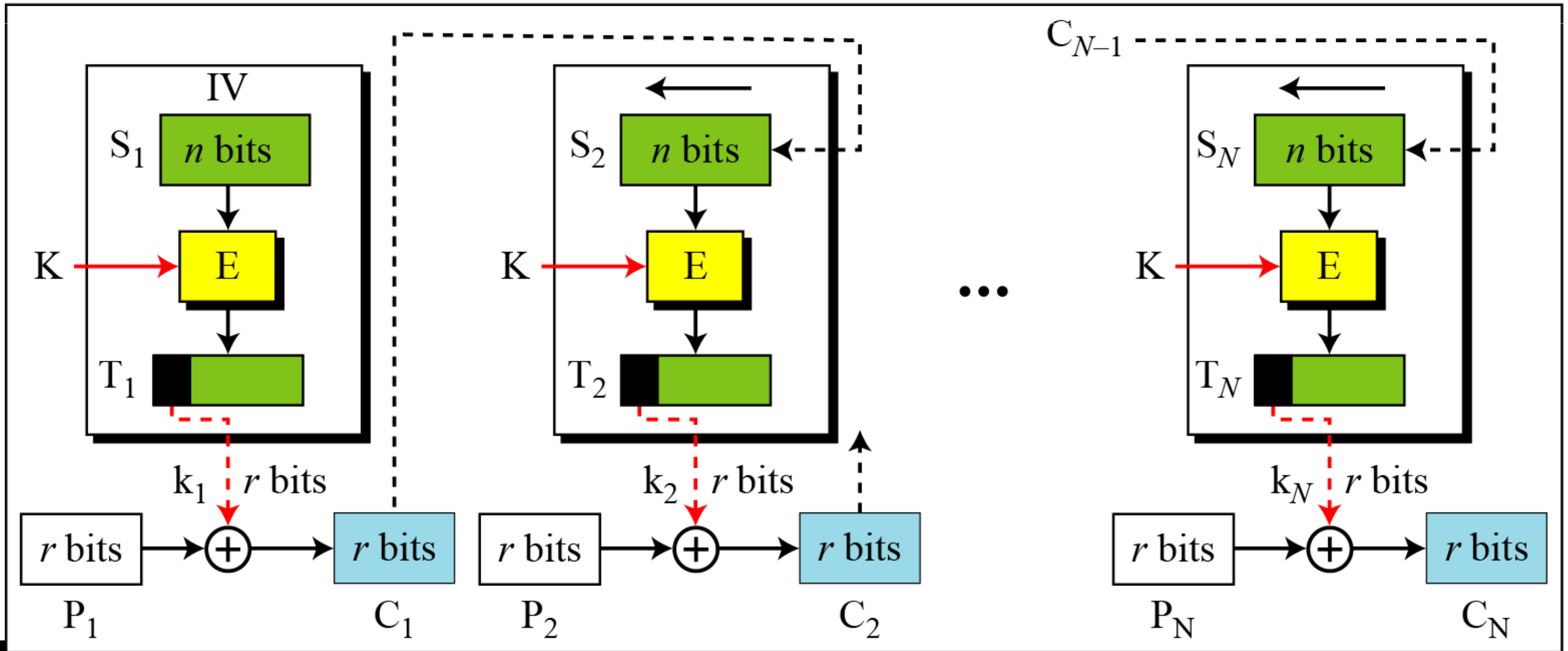
Cipher Block Chaining (CBC) Mode

- Error Propagation
  - a single bit error in ciphertext block C_j during transmission may create error in most bits in plaintext block P_j during decryption.
  - What about plaintext block P_{j+1} ???

Cipher Feedback (CFB) Mode

- In situations where,
  - we need to use DES or AES as secure ciphers,
  - but the plaintext or ciphertext block sizes are to be smaller.

E : Encryption
D : Decryption
S_i: Shift register
P_i: Plaintext block i
C_i: Ciphertext block i
T_i: Temporary register
K: Secret key
IV: Initial vector (S_1)



Encryption

Cipher Feedback (CFB) Mode...

- The relation between plaintext and ciphertext blocks is shown below:

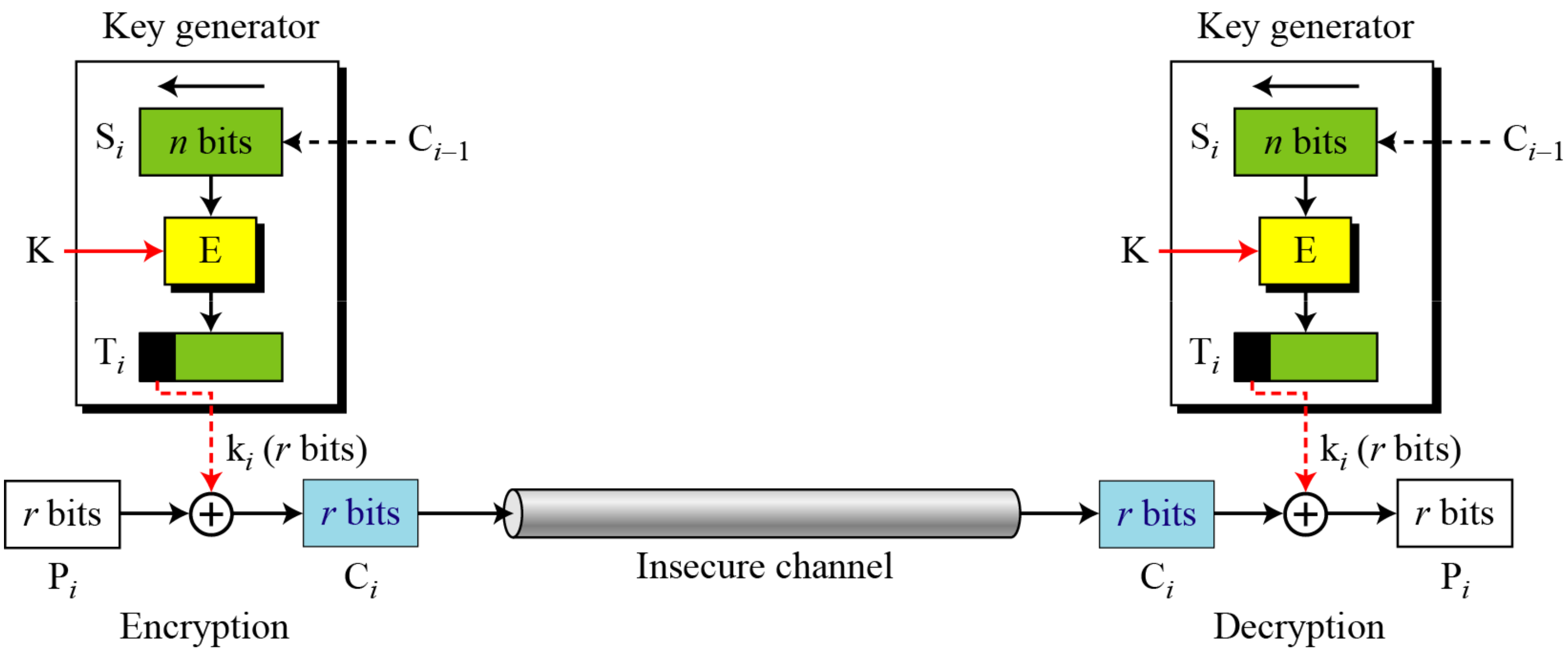
Encryption: $C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$
Decryption: $P_i = C_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

Cipher Feedback (CFB) Mode…

- Security analysis
  - No padding is required
  - System does not have to wait until it has received a large block of data before starting encryption
  - Less efficient than CBC or ECB
    - Why???

Cipher Feedback (CFB) Mode

- CFB as a Stream Cipher

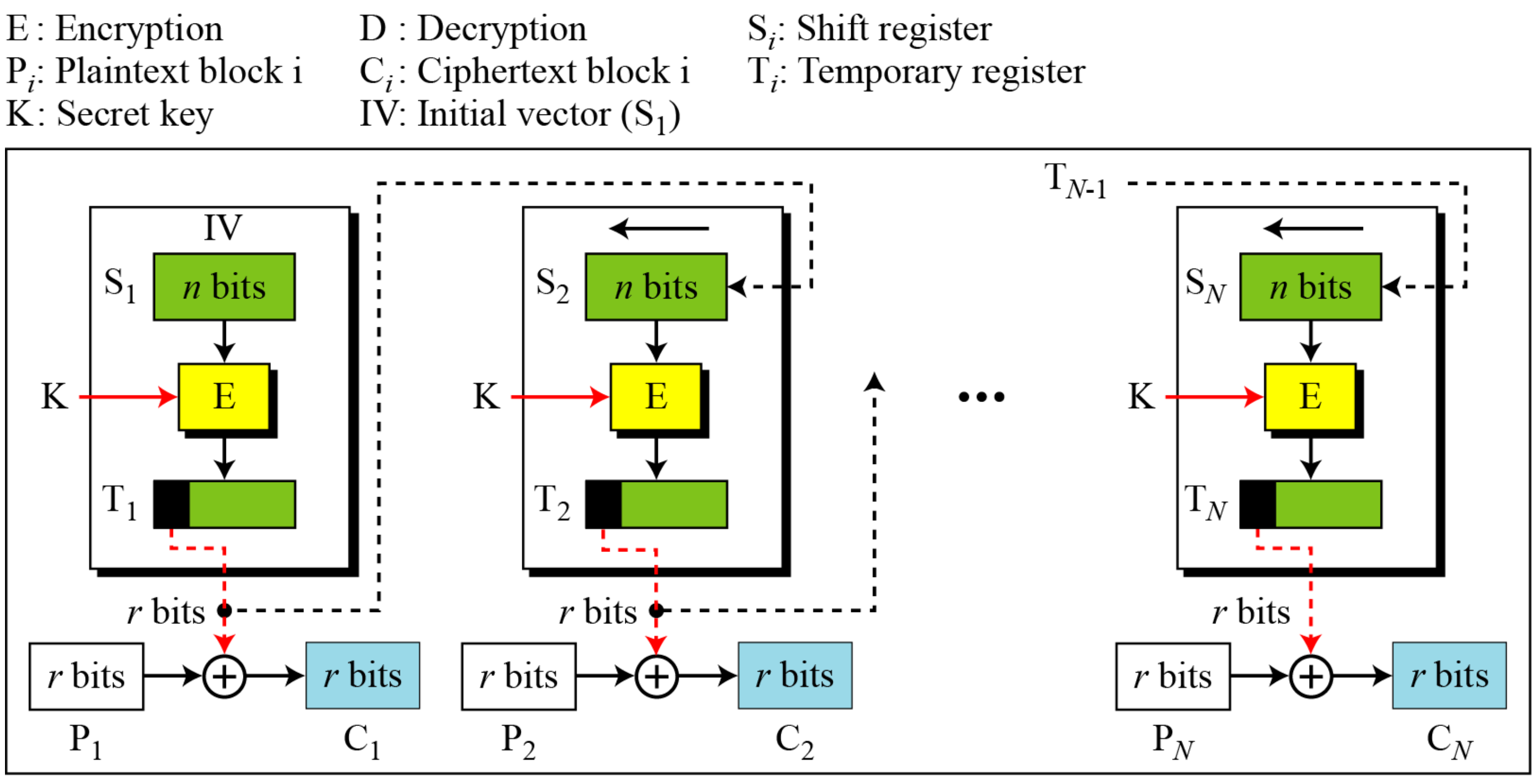


Cipher Feedback (CFB) Mode

- Error Propogation
  - What if a single bit of C_i is changed?
  - What will be the effect on P_i ?
  - What will be the effect on subsequent blocks?

Output Feedback (OFB) Mode

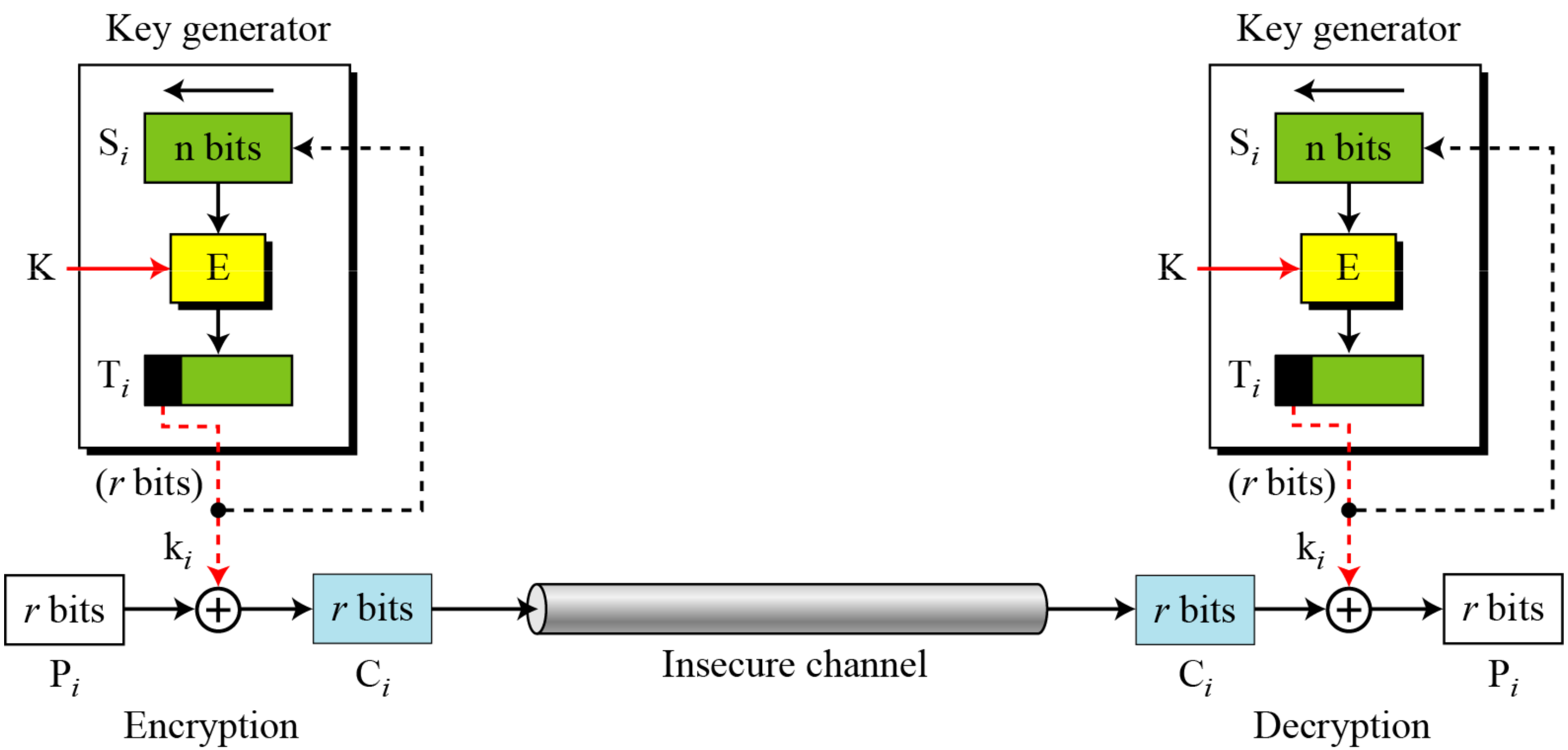
- In this mode each bit in the ciphertext is independent of the previous bit or bits.
- This avoids error propagation.



Encryption

Output Feedback (OFB) Mode...

- OFB as a Stream Cipher



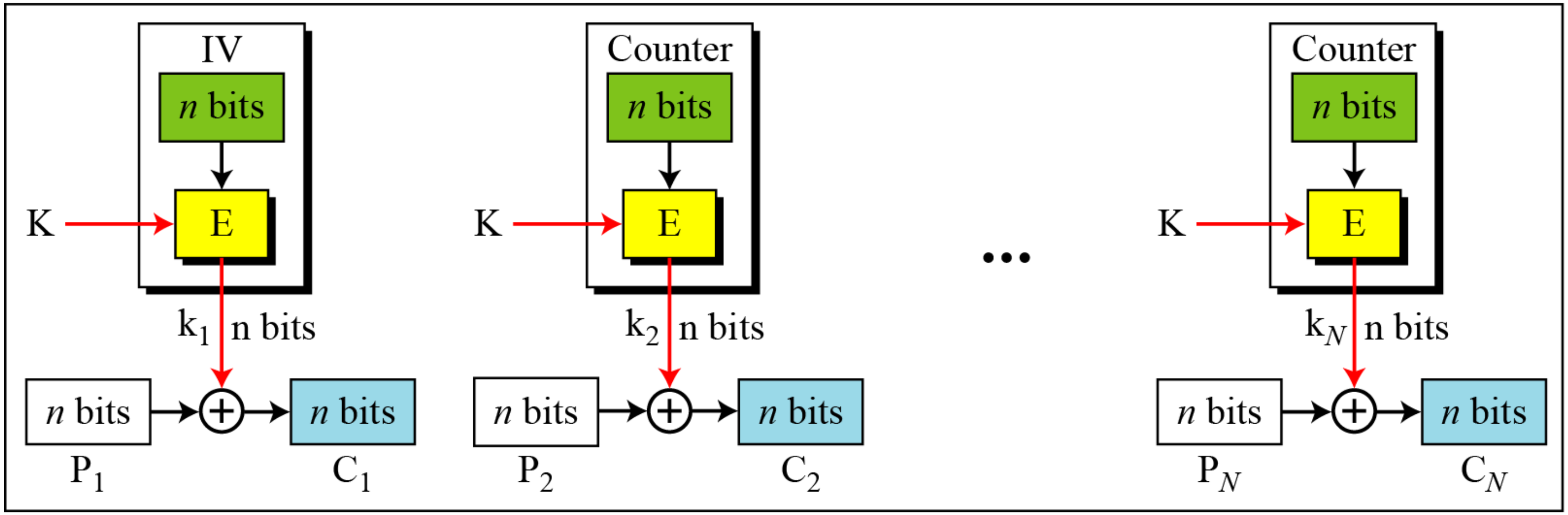
Counter (CTR) Mode

- In the counter (CTR) mode, there is no feedback.
- The pseudorandomness in the key stream is achieved using a counter.

E : Encryption
P_i : Plaintext block i
K : Secret key

IV: Initialization vector
C_i : Ciphertext block i
k_i : Encryption key i

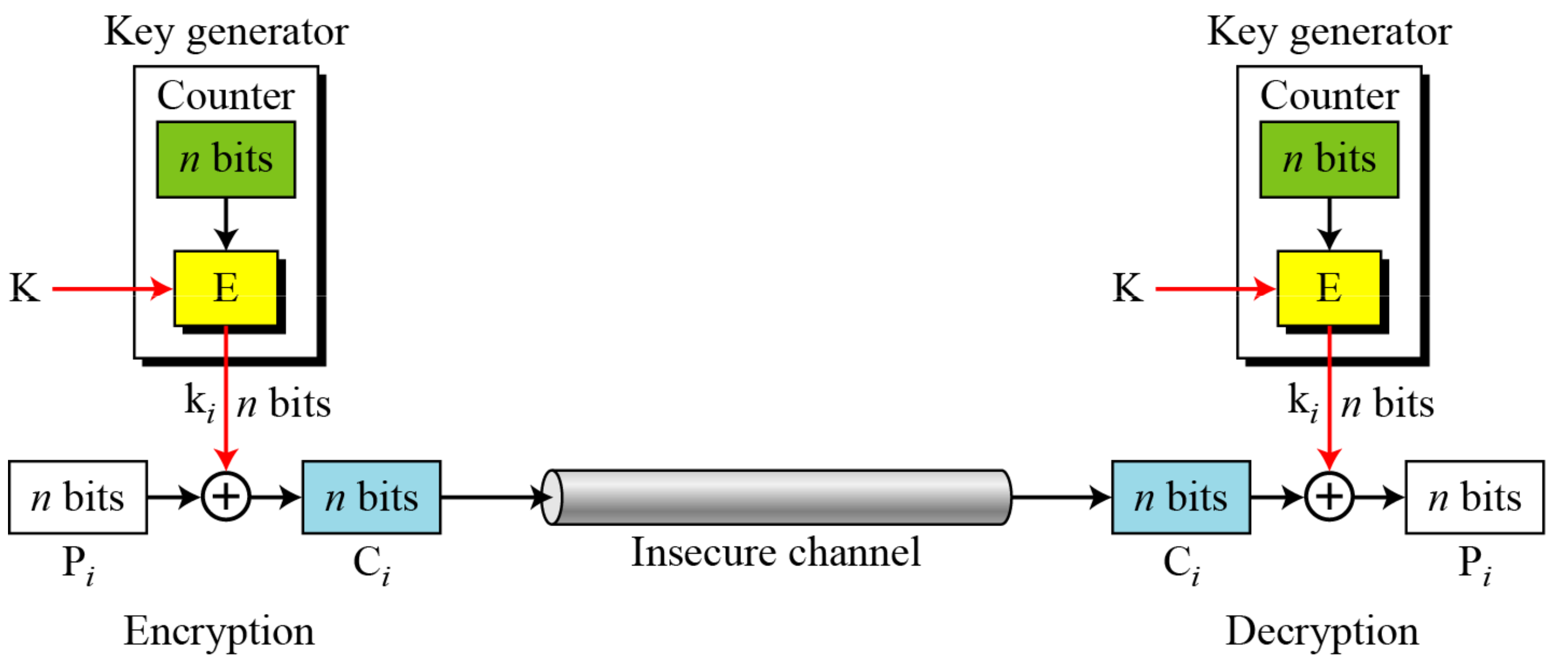
The counter is incremented for each block.



Encryption

Counter (CTR) Mode

- CTR mode as a stream cipher



Comparison of Different Modes

Table 8.1 *Summary of operation modes*

Operation Mode	*Description*	*Type of Result*	*Data Unit Size*
ECB	Each n-bit block is encrypted independently with the same cipher key.	Block cipher	n
CBC	Same as ECB, but each block is first exclusive-ored with the previous ciphertext.	Block cipher	n
CFB	Each r-bit block is exclusive-ored with an r-bit key, which is part of previous cipher text	Stream cipher	$r \leq n$
OFB	Same as CFB, but the shift register is updated by the previous r-bit key.	Stream cipher	$r \leq n$
CTR	Same as OFB, but a counter is used instead of a shift register.	Stream cipher	n